DECUS
PROGRAM LIBRARY

DECUS NO.	8-292
TITLE	FAST FOURIER TRANSFORM AND FAST WALSH-FOURIER TRANSFORM
AUTHOR	R. G. Smith
COMPANY	Carleton University Ottawa, Ontario, Canada
DATE	August 1969
SOURCE LANGUAGE	PAL

1. IDENTIFICATION

1.1 Carleton 8-D-4

1.2 Fast Fourier Transform
and
Fast Walsh-Fourier Transform

1.3 August 1969

Carleton University
Ottawa, Ontario

R. G. Smith

2. ABSTRACT

Program 1 (FFT) computes the 512 point Energy Density Spectra of two real signals using the Fast Fourier Transform algorithm.

Program 2 (FWFT) computes the 512 point Fast Walsh-Fourier Transform of a real signal.

Fixed point arithmetic is used for all computation. A hardware bit inverter is employed for speed, and auto-ranging is used to decrease roundoff error.

3. REQUIREMENTS

3.1 Storage

(a) FFT: program: 0-777

sine/cosine arrays: 1200-2177

real data array: 2200-3177

imaginary data array: 3200-4177

(b) FWFT: program: 0-377

data array: 600-1577

3.3 Equipment

PDP-8 with EAE, A/D converter (at least two channels), Display System, External Interrupt, and Hardware Bit Inverter.

The bit inverter logic diagram (minus the IOT board) is shown in Figure 1.It inverts the order of the low order 9 bits of the accumulator in 3.75 μsec. using the following IOT:

6777 - Strobe AC into bit inverter; clear AC; strobe bit inverter contents into AC

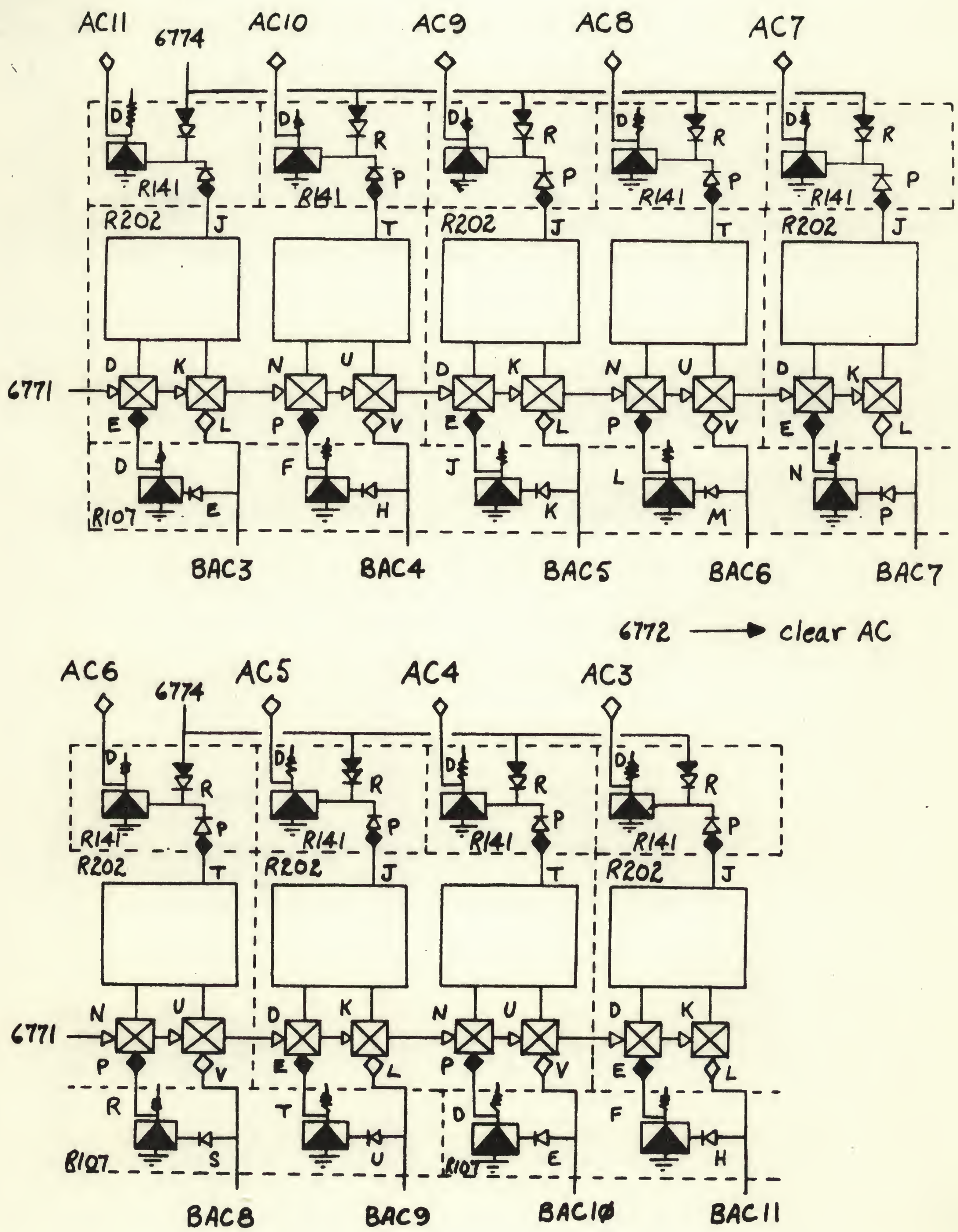

Figure 1.: Bit Inverter

The 6424 instruction seen in these programs is used to reset the external interrupt at the Carleton installation. It should be replaced by a NOP or the particular instruction in use where the programs are implemented. The interrupt is used solely to time the sampling of input data.

If it is not possible to build the bit inverter of Figure 1, the following software bit inverter may be employed. The cost is 174 μsec. per bit inversion. The number to be bit inverted is stored in INDX.

```
        TAD BITS    / NUMBER OF BITS INVOLVED
        DCA CNTR     / IN BIT INVERSION
        TAD INDX
        CLL RAL     / SHIFT LOW ORDER 9 BITS TO
        RTL     / HIGH ORDER AC
        MQL     / STORE IN MQ
LOOP,   SHL     / ROTATE HIGH ORDER BIT
        0000     / INTO LOW ORDER AC
        RAR     / SHIFT INTO LINK
        TAD INDX1     / STORE BIT INVERTED BIT
        RAR     / IN NEW POSITION IN WORD
        DCA INDX1
        ISZ CNTR     / REPEAT 9 TIMES
        JMP LOOP
        TAD INDX1
        CLL RAR     / REPLACE BIT INVERTED WORD
        RTR     / IN LOW ORDER 9 BITS OF AC
        HLT
BITS, -9
INDX, 0
INDX1, 0
CNTR, 0
```

4. USAGE

4.3 Switch Settings

FFT - if S_0 is on, the Energy Density Spectrum of the signal from channel 1 of the A/D converter is displayed. If S_0 is off, the Energy Density Spectrum of the signal from channel 0 is displayed.

4.4 Start Up

FFT: 0600 = Sampling start

0200 = FFT start

FWFT: 0200 = Sampling start

0031 = FWFT start

0257 = Inverse transform start

6. DESCRIPTION

6.3 Scaling

A pseudo-floating point procedure is used to decrease roundoff error. If all results in a given pass are less than ½ in magnitude (real and imaginary parts both less than ½), then scaling by two is deleted in the next pass. A counter (SCALE) is used to note the number of passes on which scaling down is omitted. As pointed out in (1), overflow is a slight possibility with this procedure, and it should be used with some caution. The input is scaled to ensure that all values are less than ½.

7. METHODS

7.1 Discussion

Program 1 breaks down into three segments. The first is a 512 point complex FFT program. The second is an unscrambling routine which produces the two Energy Density Spectra from the output of the complex FFT. The third segment is an Input/Output package.

Interleaved sampling of two real signals from channels 0 and 1 of the A/D converter is performed by the input package. The samples of the signal on channel 0 are stored in the real array, and the samples of the signal on channel 1 are stored in the imaginary array. Because of the interleaved sampling, the sampling frequency must be twice what is desired for the individual signals. The resultant Energy Density Spectrum of the channel 0 signal is restored in the real data array, and the Energy Density Spectrum of the channel 1 signal is restored in the imaginary data array.

The program may be adapted to perform a complex FFT of one signal through use of a suitable input routine, and deletion of the unscrambling routine (400-577).

The FWFT program produces the 512 point Walsh-Fourier transform of a real signal on channel 0 of the A/D converter. The resultant transform is stored in the data array and displayed on the CRT. The inverse Walsh-Fourier transform of the contents of the data array (in proper order) may be obtained by starting the program at 0257.

7.2 Algorithms

Program 1 (FFT) is an implementation of the tree graph of Figure 2. Use of this form enables faster operation due to a minimization of sine/cosine references. The production of spectral components in proper order also simplifies the implementation of the unscrambling routine. Bit inversion at the input makes use of the "dead" time during an A/D conversion, and therefore costs essentially nothing, in terms of processing time. The 0, $\pi/2$, $\pi/4$, and $3\pi/4$ cases are treated specially, resulting in a significant savings in the number of complex operations necessary in the evaluation of the FFT.

The unscrambling routine exploits the conjugate symmetry of the transform of real data in order to reduce the computational requirements. If two real data sequences are combined to form a complex sequence with one sequence constituting the real part, and the other the imaginary part, then the transforms of these real sequences, say $\{B_r\}$ and $\{C_r\}$, can be computed from the transform of the complex sequence, say $\{S_r\}$:

$$B_r = \tfrac{1}{2}(S_r + S^*_{N-r}) \qquad r=0,1,\ldots,N/2$$

$$B_{N-r} = B^*_r$$

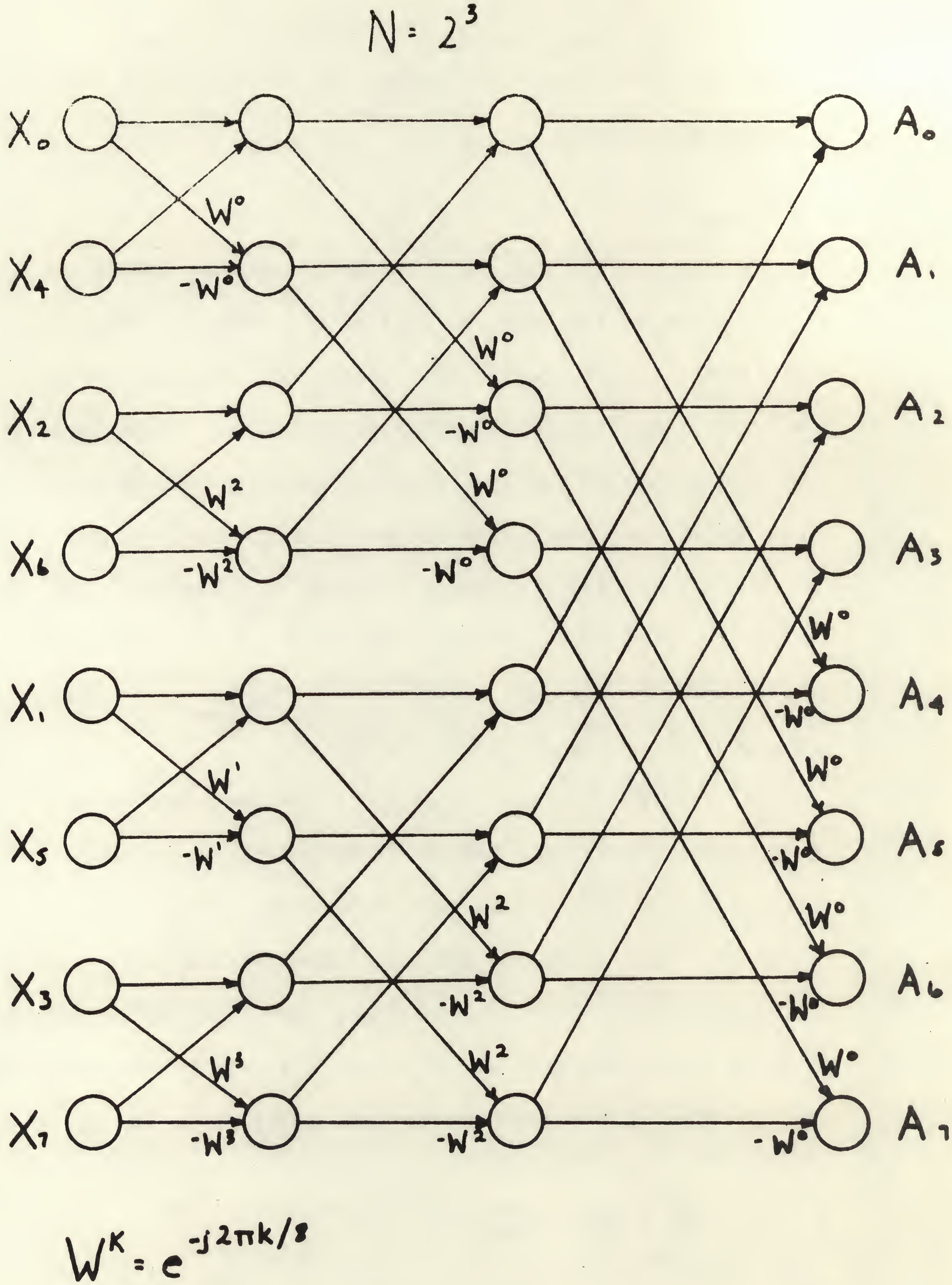

Figure 2.: FFT Tree Graph

and

$$C_r = \tfrac{1}{2}j(S^*_{N-r} - S_r)$$

$$r=0,1,\ldots,N/2$$

$$C_{N-r} = C^*_r$$

where all indices are interpreted modulo N. The unscrambling routine uses the output of the complex FFT ($\{S_r\}$) to compute $\{|B_r|^2\}$, which is restored in the real data array, and $\{|C_r|^2\}$, which is restored in the imaginary data array.

The program can be adapted for use with N a power of two less than 9 by making the following changes:

(i) NUM becomes the new number of points

(ii) NEND becomes minus the new number of passes in the complex FFT.

(iii) The bit inverted numbers produced by the bit inverter must be shifted 2^{9-p} places to the right, where the new number of points is $N=2^p$.

The sine/cosine arrays need not be changed, because the values are stored in bit inverted order.

The FWFT program is an implementation of the tree graph of Figure 3. This is a rearranged version of the graph presented by Shanks (2). It is exactly the same as the complex FFT part of program 1, with the multiplications removed.

For further discussion of the methods used, the reader is referred to Smith (1).

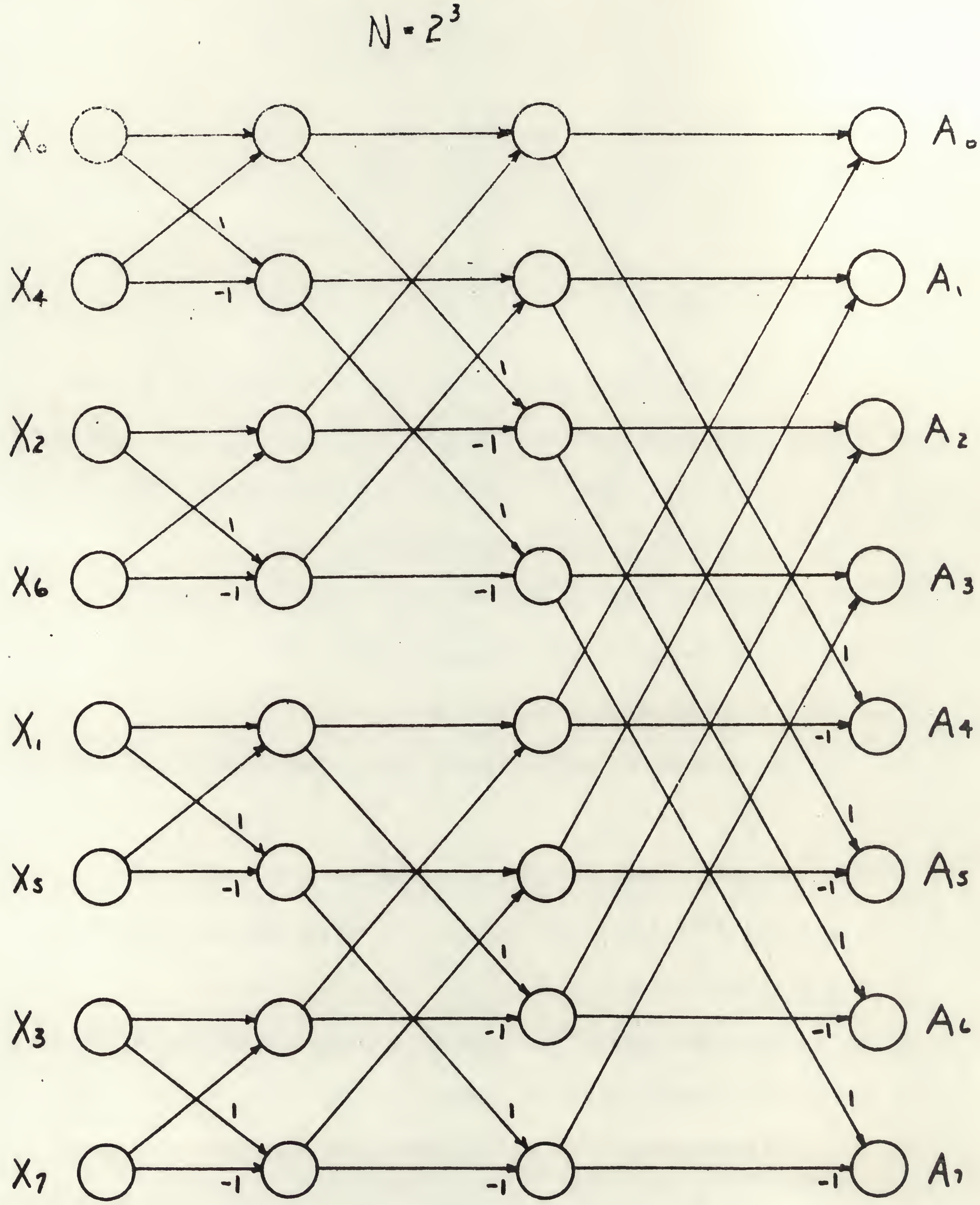

Figure 3.: FWFT Tree Graph

8. FORMAT

8.4 Miscellaneous

A new number system is employed in these programs. It has been found to be approximately 20% faster for binary multiplications and 30% faster for binary add and scale operations than the normal two's complement number system. All numbers are represented as positive quantities, so that no signing is necessary in binary multiplications. Numbers in this system can be converted to the normal two's complement number system by adding of 4000_8, and vice versa.

In this system, the number a, $-1 \leq a < 1$, is represented as $1 + a + 2n = 1 + a$, where n is any integer. The following operations are defined , and two's complement arithmetic is used:

(i) $-(1 + a) = 1 - a$

(ii) $\dfrac{(1 + a) + (1 + b)}{2} = 1 + \dfrac{(a + b)}{2}$

(iii) $\dfrac{(1 + a) - (1 + b)}{2} = 1 + \dfrac{(a - b)}{2}$

(iv) $(1 + a)(1 + b) - (1 + a) - (1 + b) = 1 + ab$

(v) $(1 + a) + (1 + b) + 1 = 1 + (a + b)$

where (1+a), and (1+b) are numbers in the system.Use of the operations is evident in the programs. It can be seen that this number system is directly compatible with both A/D and D/A converters. For further discussion, the reader is again referred to (1).

9. EXECUTION TIME

9.3 Average

FFT - .933 sec.

FWFT - .217 sec.

9.4 Minimum

FFT - .675 sec.

FWFT - .217 sec.

10. REFERENCES

(1) R. G. Smith, "the Fast Fourier Transform and the Small Computer", M. Eng. Thesis, Carleton University, Ottawa, Ontario, 1969.

(2) J. L. Shanks, "Computation of the Fast Walsh-Fourier Transform", IEEE Trans. on Computers, Vol. C-18, No. 5, pp. 457-459, May 1969.